IN THE CLAIMS

Please amend the claims as follows:

Claim 1 (Previously presented): A method of operating a computer having a pipelined processor having a Branch Target Buffer (BTB) table comprising:

creating a recent entry queue comprising a small subset of entries in said BTB table logically positioned in parallel with said BTB table;

organizing said recent entry queue as a First In First Out (FIFO) queue wherein when a previously presented entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said previously presented entry;

organizing said BTB table with a plurality of multi-associative classes with said recent entry queue being associative;

defining said recent entry queue logically as a subset of said BTB table and coupled to track a last number of branches entered into said BTB table;

comparing each previously presented entry to most recent entries in said recent entry queue ; and

blocking duplicate entries from being installed into said BTB table and said recent entry queue by examining contents of said recent entry queue for duplicate entries prior to a write into said BTB table and into said recent entry queue; and in addition allowing decoding to be delayed by a defined number of cycles such that a branch of interest can be delayed from decoding in order to allow a given entry in said BTB table to be detected in time for future decoding of said branch of interest.

Claims 2-7 (Canceled)

Claim 8 (Previously presented): The method of claim 1 comprising searching said BTB table for a next predicted branch and evaluating said recent entry queue while said BTB table is being indexed.

POU920030068US1

Claim 9 (Previously presented): The method of claim 8 wherein:

said recent entry queue maintains a depth up to the associativity of said BTB table;

whereby while said BTB table is indexed, said recent entry queue positions are input to comparison logic.

Claim 10 (Previously presented): The method of claim 8 comprising searching said recent entry queue for a matching branch in parallel to searching BTB table output.

Claim 11 (Previously presented): The method of claim 10 comprising creating hit detect logic to support the associativity of said BTB table.

Claim 12 (Previously presented): The method of claim 8 comprising using a subset of the recent entry queue as a subset of said BTB table.

Claim 13 (Previously presented): The method of claim 12 comprising fast indexing of recently encountered branches.

Claim 14 (Previously presented): The method of claim 12 comprising:

providing a complete recent entry queue; and

searching said complete recent entry queue to block duplicate BTB table writes.

Claims 15 -20 (Canceled)

Claim 21 (Previously presented): The method of claim 1 comprising staging writes to said BTB table in said recent entry queue.

Claim 22 (Previously presented): The method of claim 21 comprising delaying a write and placing said write in said recent event queue.

Claim 23 (Previously presented): The method of claim 22 comprising detecting a predicted branch while a BTB write is temporarily staged in said recent entry queue.

- 8 -

Claim 24 (Currently Amended): A ~~computer having a pipelined processor~~ system comprising:

 a computer having a pipelined processor, said system being capable of performing a method comprising:

 ~~a comparator for~~ comparing a Branch Target Buffer (BTB) table with a recent entry queue [[;]] with said recent entry queue comprising a set of BTB table entries logically positioned in parallel with said BTB table;

 ~~said computer~~ organizing said recent entry queue as a FIFO queue wherein when a previously presented entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said previously presented entry;

 organizing said BTB table ~~being organized~~ into multi-associative classes and said recent entry queue being associative;

 logically defining said recent entry queue being ~~logically defined~~ as a subset of entries in said BTB table and coupled to track a last number of branches entered into said BTB table;

 comparing each previously presented entry to most recent entries into said recent entry queue; and

 said recent entry queue blocking duplicate entries from being installed into said BTB table and into said recent entry queue by examining the contents of said recent entry queue for duplicate entries prior to a write into said BTB table and said recent entry queue and in addition allowing a decode to be delayed by a defined number of cycles such that a branch of interest can be delayed from decoding in order to allow a given entry in said BTB table to be detected in time for future decoding of said branch of interest.

Claims 25- 26 (Canceled)

Claim 27 (Currently Amended): The system ~~computer~~ of claim 24 wherein said recent entry queue is fully associative for reading.

Claim 28 (Currently Amended): A program product for creating a recent entry queue in a pipelined processor, said recent entry queue comprising comprising a computer readable medium having computer readable code thereon for controlling and configuring a computer having a pipelined processor and a Branch Target Buffer (BTB) table, to create a recent entry queue; said recent entry queue said BTB table comprising a set of BTB table entries logically positioned in parallel with said BTB table comprising;

computer readable storage media embodying computer readable program code for performing a method comprising:

organizing said recent entry queue as a FIFO queue wherein when a previously presented entry is placed into said recent entry queue, an oldest entry therein is moved out to make room for said previously presented entry;

organizing said BTB table into a plurality of associative classes and said recent entry queue being associative;

defining said recent entry queue being logically defined as a subset of said BTB table and coupled to track a last number of branches entered into said BTB table;

comparing each previously presented entry to most recent entries into said recent entry queue;

blocking duplicate entries from being installed into said BTB table and said recent entry queue by examining contents of said recent entry queue for such duplicate entries prior to a write into said BTB table and said recent entry queue and in addition for allowing a decode to be delayed by a defined number of cycles such that a branch of interest can be delayed from decoding in order to allow a given entry in said BTB table to be detected in time for future decoding of said branch of interest.

Claims 29-30 (Canceled)

POU920030068US1

Claim 31 (Previously presented): The program product of claim 28 comprising code for making said recent entry queue fully associative for reading.

Claims 32- 39 (Canceled)

Claim 40 (Currently amended): The program product of claim 28 comprising code for using a subset of the recent entry queue as a subset of the BTB [[.]] and comprising code for fast indexing recently encountered branches.

Claim 41 (Currently amended): The program product of claim 28 comprising code for using a subset of the recent entry queue as a subset of the BTB [[.]] and comprising code for searching a complete recent entry queue to block duplicate BTB table writes.

Claim 42-46 (Canceled)

Claim 47 (Previously presented): The program product of claim 58 comprising code for delaying decoding until said BTB table predicts a branch.

Claim 48 (Previously presented): The program product of claim 28 comprising code for staging writes to said BTB table in said recent entry queue.

Claim 49 (Previously presented): The program product of claim 48 comprising code for delaying a write and placing said write in said recent event queue.

Claim 50 (Previously presented): The program product of claim 49 comprising code for detecting a predicted branch while a BTB write thereof is temporarily staged in the recent entry queue.

POU920030068US1

Claim 51 (previously presented): The method of claim 1 wherein said recent entry queue is fully associative for reading.

Claim 52 (previously presented): The method of claim 1 comprising delaying decoding until after a fixed number of cycles.

Claim 53 (previously presented): The method of claim 1 comprising delaying decoding until said BTB table predicts a branch.

Claim 54 (previously presented): The program product of claim 28 further comprising code for organizing said recent entry queue as a FIFO queue.

Claim 55 (previously presented): The program product of claim 28 further comprising code for writing an entry into said recent entry queue when an entry is written into said BTB table.

Claim 56 (previously presented): The program product of claim 28 comprising code for creating hit detect logic to support associativity of said BTB table.

Claim 57 (previously presented): The program product of claim 28 comprising code for using a subset of said recent entry queue as a subset of said BTB table.

Claim 58 (previously presented): The program product of claim 28 comprising code for delaying decoding until after a fixed number of cycles.

Claim 59 (previously presented): The program product of claim 58 comprising code for delaying decoding until said BTB table predicts a branch.

POU920030068US1